AD-A196 451

DTIC FILE COPY

2

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR- 88-0616
6a. NAME OF PERFORMING ORGANIZATION: Honeywell Inc; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: AFOSR/NE
6c. ADDRESS (City, State and ZIP Code): 10701 Lyndale Ave So, Bloomington, MN 55420	7b. ADDRESS (City, State and ZIP Code): Bldg 410, Bolling AFB, DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: SAME AS 7A; 8b. OFFICE SYMBOL (If applicable): NE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F49620-86-C-0082
8c. ADDRESS (City, State and ZIP Code): Same AS 7b	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	DARPA 5794	01	

11. TITLE (Include Security Classification): OPTICAL SYMBOLIC PROCESSOR FOR EXPERT SYSTEM EXECUTION

12. PERSONAL AUTHOR(S): Professor Husain

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Quarter Technical	FROM 01Dec87 TO 29Feb88		

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The efforts in this quarter were in the design and analysis of electronic shuffle-exchange networks (SENs), the identification of the bottlenecks in the performance of the network, and the subsequent determination of the guidlines for implementing and optical SEN such that optics can address the electronic limitations and provide the capability of designing a high-performance network.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
GILES	(202) 767-4933	NE

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

OPTICAL SYMBOLIC PROCESSOR FOR EXPERT SYSTEM EXECUTION
QUARTERLY TECHNICAL REPORT

December 1, 1987 to February 29, 1988

Sponsored by
Air Force Office of Scientific Research
and
Advanced Research Projects Agency (DOD)
ARPA Order No. 5794
Contract #F49620-86-C-0082

Prepared by

Aloke Guha

Honeywell Corporate Systems Development Division
Honeywell Sensors and Signal Processing Laboratory

Submitted by: _____
Aloke Guha, Principal Investigator

Approved by: _____
Anis Husain, Section Head

Approved by: _____
Ben Hocker, Department Manager

SUMMARY

The goal of this program is to develop a concept for an optical computer for symbolic computing by defining a computational model of a high level language, examining the possible devices for the ultimate construction of a processor, and by defining the required optical operations.

The efforts in this quarter were in the design and analysis of electronic shuffle-exchange networks (SENs), the identification of the bottlenecks in the performance of the network, and the subsequent determination of the guidelines for implementing an optical SEN such that optics can address the elctronic limitations and provide the capability of designing a high-performance network.

The analysis of electronic SENs required designing the interface between the processors and the SEN, the smart exchange switch, and means of laying out the perfect shuffle within the board. We considered both GaAs and ECL technologies to determine the highest performance of an electronic SEN. Our results show that when a large number (1024 or more) of specialized graph reduction SPARO processors, whose complexity and sizes were estimated on paper, are packed on a board for high speed parallel computing, there is a severe performance degradation due to the limited parallelism in transferring messages. Our current focus has therefore been directed to using optics for implementing a high-bandwidth and high-density SEN at the board level. We are also completing the requirements of the optical smart exchange switch, and assessing its feasibility.

The requirement of high density I/O for boards is not unique to SENs. A formal analysis of board I/O requirements in transferring messages in parallel between PEs was conducted to compare SENs, hypercubes, and crossbars. Our results reveals that if a large number of boards are used in implementing the architecture, then a single-stage SEN is the best choice as long as the network load is not very high.

The analysis of the electronic SENs has provided the requirements for designing the optical SEN. Two aspects of the optical SEN are under investigation. First, implement an optical shuffle across multiple boards of PEs for parallel data transfers between processors. Second, design and evaluate the smart exchange switch in optics. The critical component for a high-performance SEN is the high bandwidth interboard shuffle implementation since that problem appears to be the primary bottleneck.

The tasks slated for completion in the next quarter are the determination of the optical interboard shuffle and the design of the optical smart exchange switch. We also expect to outline the experimental plans necessary for demonstrating the board-level shuffle in Phase III.

1. SEN AND PROCESSOR ARRAY

The general scheme for the single-stage shuffle-exchange network (SEN) is shown below in Figure 1. The Processor Array, consisting of 1024 processing elements (PEs) for fine-grained computing, communicates with the network through the Network Interface (NI), or the control portion of the network that handles the transfer of messages between the PEs and the network. Here we will focus on the complexity and the cycle time, the delay experienced by a message to pass once through the shuffle-exchange stage, of the network.

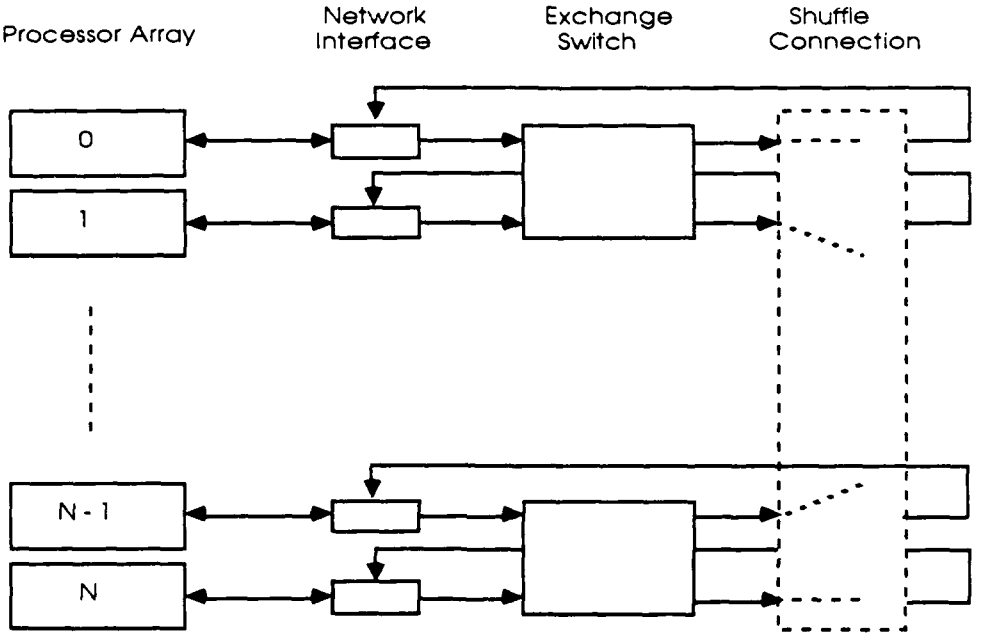

Figure 1. SEN and Processor Array Interface

We will assume that the PEs are implemented electronically since our previous studies concluded that optical implementations of even simple specialized graph reduction processors are not currently feasible. The network, however, may benefit from an optical implementation. Here we will therefore examine the implementation of the network in electronics, hybrid electrooptics, and optics to assess their relative merits and establish what network architectures are most cost-effective.

To isolate the peformance of the SEN from that of the implementation of the PEs, we will define the network cycle to be the difference of the time when a message is accepted into the NI and the time when it is loaded back into NI for delivery or for recirculation.

Each PE, as Figure 2 shows, contains two buffers for storing outgoing and ingoing messages. A message to be sent to another PE is queued at the output buffer (OB). At the beginning of a network cycle (defined as the time taken by a message to cycle once through the shuffle-exchange network), if there is a message in the OB, the PE requests access into the network through the handshake line Processor Request. The

network can receive a message if there is no circulating message at the PE. The NI corresponding to the PE communicates this information to the PE via the Processor Access line. When access is granted to the PE, the message is loaded from the OB into the NI through the message data line/lines. Similarly, when the NI at the end of a network cycle has a message to be delivered to the PE, it uses the handshake signal Network Request to check if the Input Buffer (IB) of the PE is not full. If IB is not full, the PE uses the Network Acess line to signal the NI to transfer the message over the data lines. Note in Figure 2 we have assumed that the data lines between the PEs and the NI are bidirectional ports. This has been done to reduce the number of I/O connections.

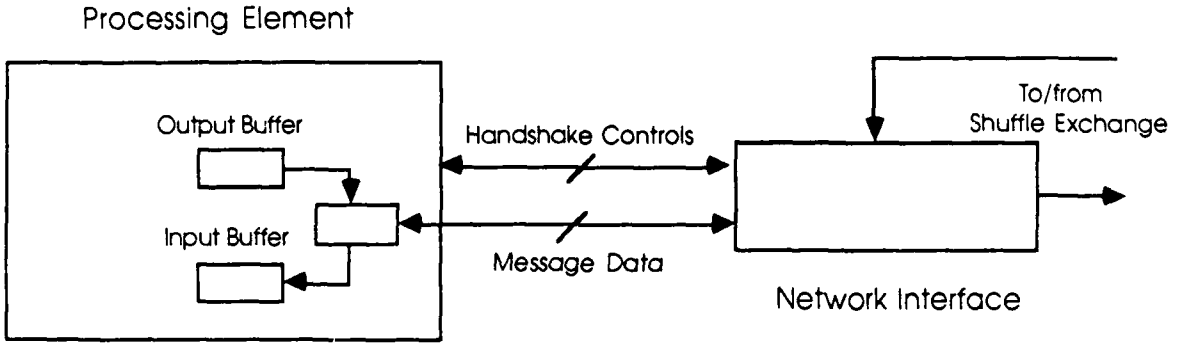

Figure 2. PE and Message Buffer interface

The mechanism for delivering messages from the SEN to the PEs proceeds as follows. When a circulating message is received in the NI, it is checked to see if it has completed $\log_2 N$ passes successfully. If it has, then it will be transferred to the PE, otherwise it reenters the SEN. This checking can be done within the NI module or inside the SE stage.

We now focus on the Shuffle-Exchange (SE) part of the network independent of the network interface portion.

1.2. Shuffle Exchange Stage

We will examine the perfect shuffle connection and the exchange switch separately. First, we outline the shuffle connection properties and then elaborate on the requirements of the exchange switch.

The nature of the connection realized by the perfect shuffle is illustrated by the SEN for 8 PEs shown below in Figure 3. The perfect shuffle permutation S for N elements is defined as:

$$S(i) = (2i + \lfloor 2i/N \rfloor) \bmod N$$

where S(i) ($0 <= i <= N - 1$) denotes the PE to which the ith PE is connected.

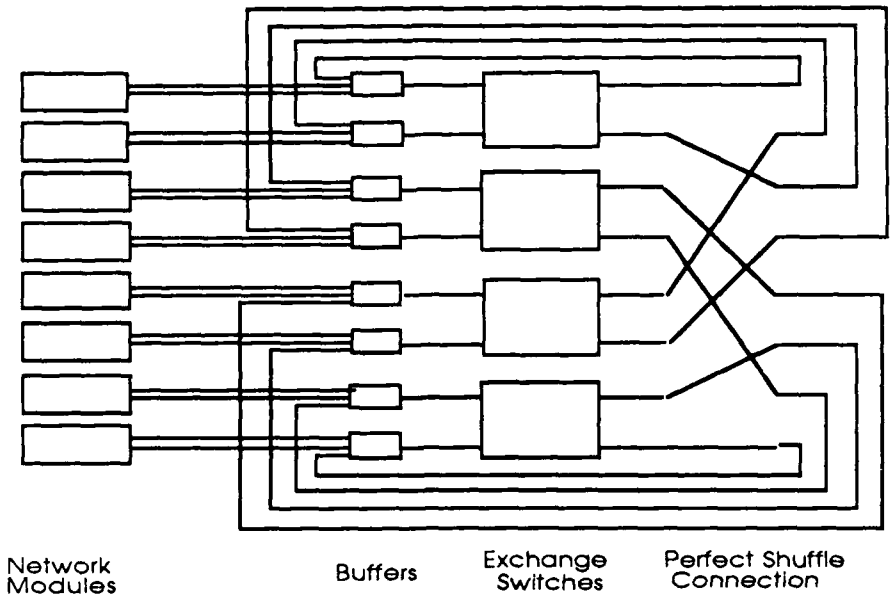

Figure 3. SEN for 8 PEs

The exchange switch within the network has many functional requirements. These include:

- means of resolving conflicts
- setting the exchange switch configuration
- updating the message control information (the number of successful passes by the message).

It must be noted that the exchange switch used for interconnecting processors are different from those used for sorting networks such as AT&T's Starlite. In a sorting network, the switch does a compare-and-exchange based on the values of the two data elements. There is usually no problem of conflict in setting the switch configuration. In a computer interconnection network, the message is routed based on the value of an address bit. (In fact, one address bit is used to set the switch during each pass of the network: the least significant bit for the first pass,..., and the most significant bit for the last pass.) Since the destination address determines the switch setting, conflicts can arise and must be resolved. The preferred method of conflict resolution is to set the switch according to the message which is closer to its destination. This scheme has been shown to be more effective than randomly resolving the conflicts. The losing message is reintroduced into the network.

The basic exchange switch, shown in the Figure 4, is one that is controlled by one input that produces a cross or bar connection between the input and the output. An useful exchange switch must satisfy all functional rquirements listed above. We will call such a switch the smart exchange switch (SES). The detailed functionality for the SES can be described by Boolean equations in terms of a number of variables described below. Although they have been derived in the previous quarterly report, we include these results here to provide a better explaination of the smart exchange switch problem.

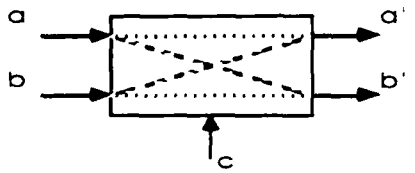

Figure 4. Basic Exchange Switch

Each input to the SES can be recognized by some header information so as to distinguish a valid message packet from noise. We can use a one-bit header, denoted by P, which we call the presence bit. A high value of P signifies the presence of a valid message packet. Each message packet has an associated counter or a mask (the reason for calling it a mask is explained below) that signifies the number of successful passes it has made through the SEN. The mask value will be denoted by M, represented modulo 1 in electronic SENs for easy decoding. Since the mask has a decoded output, the bit-wise AND of the mask and the binary address will yield the bit that determines the switch setting for the message. If the two inputs to the SES be denoted by suffixes 1 and 2, then the control C can be described by the truth table given below. Note that M represents the mask comparison ($M_1 >= M_2$) operation. R_1 and R_2 represent the mask reset controls that are necessary to handle conflicts.

	P_1	P_2	M	A_1	A_2	C	R_1	R_2
No Packets	0	0	-	-	-	-	0	0
One packet	1	0	-	0	-	0	0	0
	1	0	-	1	-	1	0	0
	0	1	-	-	0	1	0	0
	0	1	-	-	1	0	0	0
Two packets: No conflict	1	1	-	1	0	1	0	0
	1	1	-	0	1	0	0	0
Conflict	1	1	1	0	0	0	0	1
	1	1	0	0	0	1	1	0
	1	1	1	1	1	1	0	1
	1	1	0	1	1	0	1	0

Table 1. Truth table for control of SES

Using the truth table, the reduced expression for C is:

$$C = P_1 \neg P_2 A_1 + \neg P_1 P_2 \neg A_2 + P_1 A_1 \neg A_2 + P_2 \neg A_2 \neg M + P_1 A_1 M$$

where ¬ represents a NOT operation. The above expression assumes that both the deciding address bit extraction as well as the mask comparison has already been completed. There are a number of possible hardware schemes, serial and parallel, for realizing both operations.

Since the SES has the necessary information, it can be made responsible for updating the mask values, M_1 and M_2. If the mask value reaches the value $\log_2 N$, having started at 1, then message must be delivered to the processor and not allowed to reenter the network. If the mask value is less than $\log_2 N$, then the mask is either reset to 1 if it loses in conflict resolution or it is incremented modulo 1. The truth table shows four combinations of control inputs for which the mask values must be reset. The expressions for the reset signals are:

$$R_1 = P_1 P_2 \neg M \ (\neg A_1 \neg A_2 + A_1 A_2)$$
$$R_2 = P_1 P_2 M (\neg A_1 \neg A_2 + A_1 A_2)$$

2. ELECTRONIC SEN IMPLEMENTATIONS

In order to uncover the critical architectural issues in designing optical SENs, we first consider a purely electronic design. A number of issues determine the nature of the electronic implementation in terms of the size and complexity as well as performance. These include pin constraints of a chip, the number of backplane interconnects between printed circuit boards or the backplane wiring constraints, and the off-chip and off-board interconnect delay as compared to the gate delays within a chip. We examine these issues as inputs to our design methodology, and then present the possible performance and size of electronic implementations.

2.1 Pin Constraints

The design approach most affected by the limits on pin counts is the issue of parallel versus serial message data transfer. Consider the parameters of the SPARO (Symbolic Processing Architecture in Optics) architecture. There are 1024 PEs communicating via messages composed of five essential fields: the destination PE address, the source PE address, two data/address operands, and the instruction. For a 1024 processor architecture, the address is 10 bits wide. Although the specific application will decide the size of the data used, we assume that a 32-bit wide data would suffice. The instruction word was assumed earlier to be encoded as 1 bit per macro-instruction to simplify the instruction decoding in optics. In the first design only about 20 macroinstructions were assumed, requiring 20 bits for the optical instruction width. We now expect that more than 30 macroinstructions, each of which requires 2 to 3 machine cycles, will be necessary. In the case of an electronic design where binary decoding can be done relatively simply in the PE, we will assume 6 bits are required to encode all macro-instructions. The total width of the message is thus 90 bits (2(32) + 2(10) + 6). A message of this length presupposes that the mask for each message is generated in the network. However, to avoid increasing the complexity of network, it is preferrable that the mask be generated by the processor and sent as part of the message for purposes of routing in the SEN (Figure 6). The total message length would then be 100 bits. This is the nominal length of the message. Larger or smaller sizes of the messages are possible depending on the size of the data necessary. As we shall see later, the length of the message has a profound effect on the performance and implementation of the SEN.

Destination	Mask	Source	Instruction	Operand 1	Operand 2
10	10	10	6	32	32

Figure 5. SPARO message format

The control lines required between each PE and the corresponding row of the SE were discussed earlier. Four handshake lines are required: Processor Request by the PE to transfer a message from the OB of PE into the NI, Processor Grant to grant the

processor request, Network Request by the network for message delivery from the NI into IB of PE, and Network Access to grant the network request. Since messages are all of fixed length, we do not require acknowledge signals after message transfers have been completed.

The problem of pin limitations arises when considering how the message has to be transferred between the PEs and the SEN: 104 bits of control and message information if the messages are transferred in parallel between the NI and the PE, or 5 bits for control and serial data lines. Note that in our initial electronic designs all message lines are considered to be bidirectional to reduce the space overhead. In the first case, all messages (at the beginning or at the end of the network cycle) can be transferred in one clock cycle after handshaking. In the second case 100 clock cycles are required to transfer the message serially. While the serial option is 100 times slower, it requires 1/21th the number of pins at the output of the PE. There is thus a space-time trade off to be considered. The real question to be answered is the total space-time complexity. If 1024 PEs are placed on one board to reduce off-board delays, then the board has to accomodate 1024*104 or 106496 lines between the PE array and the SEN. By comparison, the serial transfer scheme only requires 5120 lines. How many PEs can be put on a chip or package and then on a board therefore depends on the pin limitations on the chip as well as the number of interconnection lines that can be squeezed on a single board. These issues in turn depend on the technology used to design the board.

2.2 Off-chip Interconnection Delays

Since the complete network (for 1024 or more PEs), that is, the exchange circuitry as well as the shuffle connection, requires a multi-chip (possibly multi-board) implementations, off-chip delays will be a design concern. The problem of interconnection delay becomes severe for a large perfect shuffle where exchange stages are switched at high speeds. The network delay, or the time taken by a message bit to pass around the complete SEN, depends partially on the actual interconnection length between the output of the exchange stage and the register that delivers the message to the input of the exchange stage (see Figure 3). Since the shuffle connection is not modular, this length increases with the size of the network. We will examine the relative importance of the interconnection delay when examining SENs implemented in different electronic technologies.

2.3 Backplane Wiring

The problem of backplane wiring arises if a multiboard implementation is necessary when all PEs and the corrresponding interconnects cannot fit on one board. The number of boards and the total delay would then be determined by the number of backplane interconnects possible. The maximum number of board-level interconnects depends on the technology used to construct the board. Thus, using thin film multilayer (TFML) boards allow for much faster and more dense interconnects than do standard PVC PCBs with edge connectors.

Both pin count and packaging constraints and limits on backplane wiring will

therefore determine the nature of transfer of the message, that is, how serial or how parallel. The other factors are the complexity of each SE stage. The larger the area of a single SE stage, the fewer PEs can be fitted on a chip and thus fewer chips on a single board. We will consider these factors in more detail in the next section.

2.4 High-Speed Electronic Implementation

We have examined the complexity of the circuitry required to implement the complete shuffle-exchange network in electronics. Both GaAs and Si ECL technologies were considered for high-speed implementations. Given the large size (1024) of the network, we initially considered bit serial transfer of messages. We examine the design implications for a parallel message transfer scheme later. As we will show, the nature of the message transfer in the network is critical in determining the complexity and performance of the network implementation.

The size of each exchange stage and its controls is estimated first to detemine the layout complexity and thus the total area, size, and speed of the network. The circuitry in the exchange includes the NI and its controls, the combinational logic to generate the exchange switch settings, and registers to hold the message during recirculation. Figure 6 shows a schematic of the electronic SEN. The operation of the network is now explained in more detail.

A message is accepted into the network with the destination address field entering first (as shown in Figure 5). The address and mask fields of the message are successively loaded into their respective registers, so that the deciding address bit can be extracted. The deciding address bit extraction is achieved by serially shifting the mask and address register contents and ANDing the output bits. While the mask register is being cyclically shifted, the mask comparison between masks of two messages can be done in parallel. The mask comparison result and the deciding address bits are fed to the switch and mask control logic. While the presence bits of messages have not been shown in Figure 6 to simplify the diagram, they can be derived in the NI by examining the processor request and grant lines. When the exchange switch is set, the registers are alternately emptied to serially pass their message to the shuffle stage. The message buffer holds the initial portion of the message while the remaining message portion passes through the exchange switch.

We now examine implementation of the above SE circuit in different technologies.

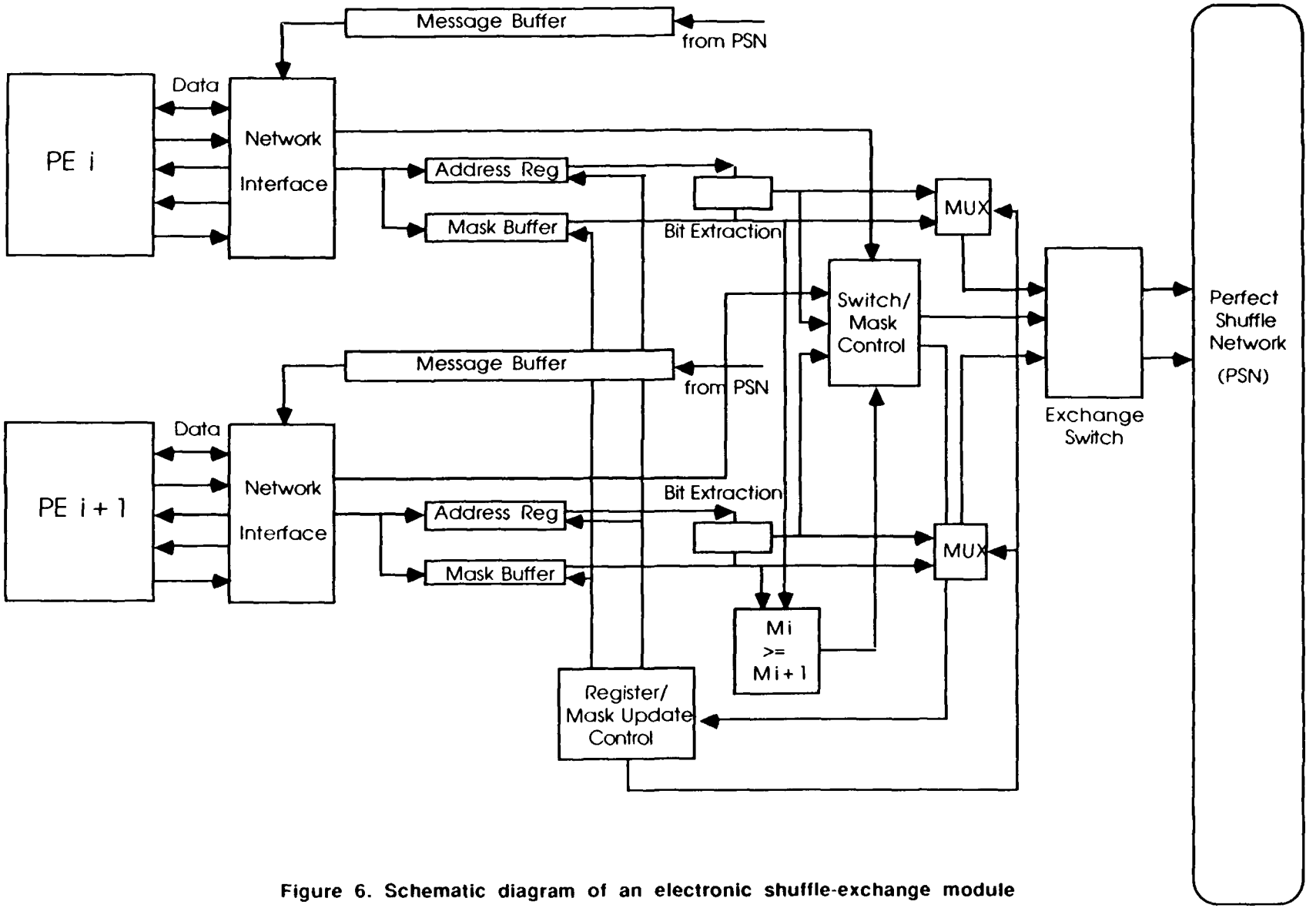

Figure 6. Schematic diagram of an electronic shuffle-exchange module

2.4.1 Network delay for GaAs implementation

In the case of GaAs, the total number of gates (the average gate is a two-input NOR or NAND) required for each exchange stage for every pair of PEs (Figure 6) is expected to be less than 1300. (The major proportion of these gates are consumed by registers and buffers.) If the fastest usable GaAs technology (Honeywell's) were employed, gates with delays of the order of 100 ps (10 GHz) could be used. Since the number levels of logic between switchable gates will be 3 to 4 on an average, the clock speed for operating the network would be about 2 GHz. At that speed, the current and near-future level of integration allows between 3000 and 4000 gates on one chip. This allows us to realistically fit in 2 exchange stages (for 4 PES) on a single chip. Using state-of-the-art multichip carriers (Honeywell's), we could fit close to 25 chips or 50 exchange stages on a rectangular MCP (multichip package) measuring 3.7 by 2.4 inches. The current limit on the number of pins in a pin grid array is about 575. Since each stage requires one output message line and five input lines, a single package can be used to pack about 100 PEs. We are assuming that the data line is bidirectional. If unidirectional data lines are used, six lines are required to connect each PE to the network.

For a 1024 SEN, we would have to put together 10 or 11 such MCPs on a PWB (printed wire board). Since standard PWBs have low dielectric constants, off-package delays will be higher than those inside the package unless polyimide substrate is used for the board. Interconnection lines would be done in copper on the polyimide. The shuffle stage would have to interconnect these 10 packages on the board. Each package would have 100 serial output lines for messages leaving the exchange stage for the shuffle connection.

Because of the non-local nature of the shuffle connection and the limit in integration, we cannot integrate the shuffle connections on the package. The off-package delays for the connections are directly proportional to the wire length. If multilayer (TFML) connections are used (currently 5 layers with 3 for ground and power), the longest vertical length between packages is 5 MCP heights, that is, 5" x 3.7" or 18.5". Since this line is large in length, there is considerable loss in the lines. It would be difficult to run GaAs at 2 GHz over long interconnection lines unless impedance matched input and output buffers are used at the input and output pins of the exchange stages. (The bigger problem is that of timing, or the distribution of clock to all parts of the SEN which is to operate synchronously.) Since the delay on copper on polyimide is 62 ps/cm, the longest interconnect delay in the shuffle stage is 2913.4 ps or nearly 3 ns. Note that the processor to exchange stage connections are not as much of a bottleneck since they are direct local connections between PEs and the MCPs for the exchange (Figure 7). We are assuming, for our preliminary analysis, that all PEs can be connected by the SEN on one board. As it turns out, this is not possible since packing 1024 PEs alone will consume a complete board-see note on processor complexity in the next section. Thus the PE array and SEN connection will be across boards and not on a single board. For purposes of determining the upper limit on the SEN performance, however, we assume that the PE and SEN connection problem can be solved, and therefore focus on the one-board SEN performance. We have assumed that load impedances will be matched at the MCP

pin boundaries. Thus the delay between connections of successive stages of the network is over 3 ns given that some gate delays have be accounted for within the package. We can assume the worst case total network delay time to be about 4 ns.

In our delay computation, we have assumed that the clock can be distributed such that no clock skews occur. At Honeywell we use a star configuration in distributing the clock within a package to ensure that the clock propagation delay is the same for all modules. If clock synchronization is a problem at the board level, and we believe it will be, we could conceivably employ a holographic optical element (HOE) to distribute the clock.

2.4.2 Network Delay for ECL Implementation

Because of the relative severity of off-chip delays in the GaAs implementation, we considered a more mature technology, ECL, as another choice for implementing the control and exchange stage of the network. The advantages of ECL over GaAs despite its slower speed is the higher level of integration as well as a relatively smaller penalty for off-chip connections. Today, we can integrate 10000 gates on an off-shelf ECL gate-array chip with relative ease. With the state-of-the-art ECL technology, maybe even 20000 gates could be put on one chip. (This would be possible since the connections in the SE stages are regular and simple with low fanouts.) We are assuming of course that the high heat dissipation problem for the on-board ECL circuitry can be solved. With such a level of integr n, one could account for SE stages for 20 PEs. To completely fit all 1024 PE con ns, we would need about 50 chips on a board. Each chip would have . ins for serially transferring data in and out of the network. This number of pins is quite feasible today. By pushing technology to its limits, the data could be clocked through at 200 MHz if delays across the board are not significant. It would be reasonable to assume that the interconnection length would be close to that of the GaAs SEN. Thus the network delay will also be close to 4 ns. We assume as before that an optical clock distribution is possible using a HOE (holographic optical interconnect) scheme. Since the ECL clock is run as much as 50 times slower than that for GaAs, the interconnection delay in the ECL SEN is a very small fraction of the cycle time.

We now examine the total cycle time of the network for each technology.

2.5 Network Cycle Time

The network cycle time is the sum of the propagation delays for the control and the exchange switch, the shuffle path delay, and the total time to transfer the message. Since the number of gates is expected to be less than 5 between clocked stages, and since the shuffle connections are compact, these delays do not slow the clock down. The message transfer is therefore dominated by the time taken to transfer 100 bits. Since the delay across the board is not significant, the head of the message will arrive through the shuffle connection to the next exchange stage before the tail of the message has passed completely through the exchange switch. Even if the two 10-bit registers are used to alternately hold 10-bit chunks of the message (see Figure 6), a message buffer is required to save a portion of the message before the exchange

switches can be set for the next pass.

In the case of GaAs, the message bits are pipelined out of the source PEs at 2 GHz. The network cycle time is 54 (100*500 ps + 4 ns) ns or effectively the network operates at 18.52 MHz. The network delay time is thus 108 times slower than the GaAs gate delay (500 ps) because of the serial transmission of the message. Since two 10-bit registers are used to hold portions of the message, a 80-bit message register is required. In case of ECL, the message transfer time is 504 (5 * 100 + 4 ns) ns. This implies that the network effectively operates at 2 MHz.

In case of either technology, the serial message transfer causes the network to be a bottleneck. This bottleneck is especially serious when the PEs operate on the same clock as the SEN. An average message requires 1.5* $\log_2 N$ network clock [Lawry and Padua] or 150 * $\log_2 N$ clock cycles to be delivered (100 bits per message), when the network load is not more than 0.25. Therefore for a 1024 SEN, a message requires 1500 clock cycles to be delivered. If a specialized reduced instruction set (for combinator graph reduction) architecture is used to implement the PE, a message can be generated at best in 3 to 5 cycles, assuming some parallel loads are allowed within the PE. Thus, the message generation rate (all 100 bits generated in parallel) is about 300 to 500 times higher than the message delivery rate if the message is transferred serially, and 3 to 5 times higher if transferred in parallel. In an ideal situation, where no bottlenecks exist, the network should deliver messages at approximately the same speed at which the PEs generate them. Thus, the message bandwidth in the network must equal the message generation bandwidth in the PE. This implies that the SEN operates on a clock that is either operating at a ridiculous 300 to 500 times faster than the PE clock when transferring messages serially, or accepts and transfers messages totally in parallel with a clock that is less than an order of magnitude faster.

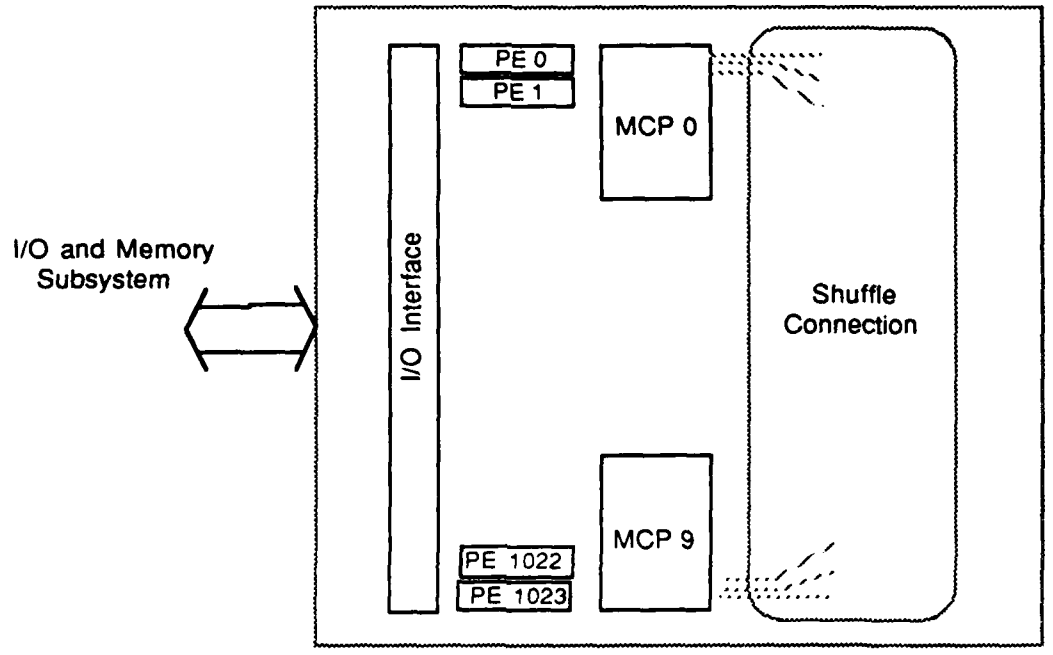

Figure 7. Scheme for an electronic SEN and PE array

2.6 Parallel versus Serial message transfers

The previous discussion showed that the message bandwidth in the network is inversely proportional to the message length. The increased bandwidth required in the SEN motivates us examine the complexity of a parallel implementation where messages are transferred in a parallel or quasi-parallel fashion.

For the same level of integration as described earlier, an ECL chip built with 20000 gates can accomodate exchange stages for 20 PEs. However, instead of 20*6 (5 between PE and exchange and 1 between output of the shuffle stage and exchange) or 120 I/O pins, now 20 * 204 (104 between PEs and exchange and 100 between output of the shuffle stage and exchange) or 4080 I/O pins are required, ignoring ground, power and clock connection pins. Since this is not possible, a single chip cannot be deemed to contain more than one exchange stage for 2 PEs since about 250 is the limit to the number of pins to a chip. In such a case, 500 such chips would have to be interconnected in a shuffle connection where each channel of the shuffle now has to connect 100 wires per channel or a total of 102400 wires for a 1024 shuffle. If board-level interconnects are used, then many boards are required in the implementation of the shuffle connection since typically only 250 to 300 backplane connections are possible with standard edge connectors. Clearly, the pin limitations and size complexity makes a large shuffle-exchange for parallel message transfer impractical in electronics.

In summary, one notes that an electronic implementation, GaAs or ECL, for a large shuffle-exchange could be operated at high speeds, over 200 MHz for ECL and over 1 GHz for GaAs. However, severe limitations of the packaging technology and the level of integration of high-speed semiconductor technology forces a serial transfer of messages when a large SEN is desired. Unfortunately, when messages are transferred serially, the message throughput varies inversely as its length. For a modest message length of 100 bits, suitable for the level of fine-grained computing, the message throughput is less than 1/100th the data rate. The following table describes the electronic SE network cycle times and the corresponding message latency. The message latency is defined, for our purposes, to be the average time required to deliver a message. We have assumed, using Lawrie and Padua's results that an a message requires an average of $1.5*\log_2 N$ cycles to deliver if the network is not loaded much beyond 0.25 (that is, about 25%of all stages have messages in transit).

Technology	Gate delay	Network latency	Message latency
GaAs	500 ps (2 GHz)	54 ns (18.52 MHz)	810 ns (1.24 MHz)
ECL	5 ns (200 MHz)	504 ns (2.0 MHz)	7.56 us (132.3 KHz)

Table 2. Performance of electronic SENs

For comparison with the above performance, we examine the same issues for electrooptic and optical implementations.

3. OPTICAL AND ELECTRO-OPTICAL SENS

To consider the possible optical implementation of the SE, we will re-list the functions required and specify optical candidates for their implementations (Table 3).

As in the electronic implementation both parallel and serial message transfers can be considered. As before a parallel message transfer scheme in optics requires that each channel be 104 (100 message lines and 4 control lines) signals wide. A total of 100K signals in a single optical system appears difficult in the near future if guided optics is used. Thus, the optical system may have to be quasi-parallel since the worst case serial transfer cannot provide acceptable throughput unless the switching speeds in the optical SEN is hundreds of times faster than the PEs. We will explore the quasi-parallel approach in more detail later.

Table 3 shows possible candidates for implementing the different required functions of the SEN in optics. Some of these ideas were presented in the last quarterly report [Guha]. Most of the components of the smart exchange switch are presently conceptual and are still under evaluation.

Since an all-optical implementation has not yet been designed, we will focus on the architecture of electrooptic and hybrid implementations. One thing is clear: if the optical network has to provide an advantage over an electronic one, using an optical serial shuffle together with an electronic exchange and control will not be an advantage. This is because the serial delay of the message in electronics is a serious bottleneck, and using an optical shuffle will only add further electron-photon conversion delays. We will therefore examine optical methods to improve the bandwidth of message transfers. Another issue that requires consideration in designing the SEN is the nature of its interface to the PE array. The network interface depends on the size of the PE and therefore on its complexity. The next subsection examines the complexity of a special-purpose graph reduction processor.

3.1 Processor Complexity

The nature of optical implementation depends on the level of connectivity, that is, whether the connections between the PEs are within the board or off-board. Since we are building a fine-grained parallel system, the size of each PE dictates the nature of connection. For this purpose, we examined the functional requirements and the architecture of a specialized combinator graph reduction (CGR) PE.

Our initial estimates show that a reduced instruction CGR PE will have about 30 to 35 hardwired instructions, each of which executes 2 to 3 steps. The typical sequence of operations are as follows. The PE receives a message, decodes it, operates on it, and sends out a message in response. If a simple ALU (no multiplier) is used, the PE could be implemented with about 1600 gates in Honeywell's high-speed (50 MHz), high density CMOS process. To give more power to the PEs, a larger ALU equipped with a mulitplier could be shared by a pair or more PEs on the same chip.

As many as 11,000 gates can be put on 1 CMOS gate array chip (400 mils X 400 mils). Thus, 6 to 7 PEs can fit on 1 chip. As many as 20 gate array chips could be squeezed on a package 3.25 " X 2.6". Thus one package would account for 120 to 140 PEs. The limit in the packaging is not in the total gates available but in the number of I/O pins. The maximum number of pins possible in such a package is 575 in a pin grid array. Thus if 100 PEs were dedicated to a package, each would have only 5 or 6 (depending on a bidirectional or unidirectional data line) I/O pins allocated to it, ignoring common clock, power, and ground lines. This limited pin allocation per PE forces a serial message transfer scheme wherein all messages in and out of the PE have to be serial.

Shuffle -Exchange Functionalities	Alternatives to Optical Implementations
Shuffle Connection Perfect Shuffle Mapping: $S(i) = (2i + \lfloor 2i/N \rfloor) \bmod N$	Bulk optics (Lohmann, Eichmann,..), or Holographic mapping
Smart Exchange Switch 1) Basic Exchange Switch: $(a', b') \leftarrow (a, b)$ where $a' = NOT(c)a + cb$ $b' = ca + NOT(c)a$ c is switch control	Birefringent substrate with grating: data intensity encoded, control polarization encoded; switch setting determined by polarization of message beam, or Polarization switching gate - gate is modulated electrooptically
2) Switch setting on Conflict Resolution	Boolean function of mask values, presence bits, address bits under mask
Representation of mask value	Spatial encoding for mask value
Representation of presence bits	Separate optical signal
Updating/ resetting mask	Spatial shift to update Reset requires separate control
3) Determining whether message should be delivered	Position detection

Table 3. Optical implementation alternatives for a SEN

At the level of integration discussed, at most 10 packages can be fit on a large board. The limit at the board level would be in the number of board connections as well. If

1024 PEs were accomodated on 1 board, the number of I/O lines in the board at 5 or 6 I/O pins/PE will be over 5000. Thus, connecting a SEN to this board is not possible using standard electronic wiring connections. More importantly, it is clear that to build a scalable machine, consisting of 10,000 PEs or more (necessary for fine-grained computing), a multiboard solution is desired. Therefore, the SEN must be operational across multiple boards and not just within the board. The number of boards required for all PEs is dependent on the functionality and granularity of a PE. We now examine this important issue in more detail.

3.2 Processor Granularity

The PE granularity influences the SEN design in two ways. First, the coarser the granularity, the fewer the PEs required to solve the problem. This implies that a small number of boards will suffice to accomodate all PEs. The architecture of the processor/memory subsystem for coarse-grained processors will of course be quite different from the one chosen here. Second, coarser-grained PEs will operate on a larger problem (that is, on subgraphs rather than on individual nodes in CGR) and therefore will have less frequent communication with other PEs. The message generation frequency will therefore be considerably lower. However, the size of messages may be considerably larger. For example, the messages may contain subgraphs rather than single node information. The increased size of messages will tend to keep the bandwidth of messages high even if the message generation rate is decreased. One way to keep the message size down to the lengths that we are considering here (100 bits) is to use a radically different architecture such as shared memory and PE clusters. Using such a different architecture implies solving a different sort of PE communication problem which we will not consider here. We will instead focus on the general tradeoff of PE granularity and message bandwidth.

In the SPARO architecture that has been developed for fine-grained CGR, each PE contains and operates on a single graph node. There is no concept of memory since the registers in the PE specify a graph node completely. While this approach seemed suitable for optics where a complex processor could not be designed, when considering an electronic implementation other problems surface. First, the number of PEs required in the architecture is not defined by the size, in terms of the number of nodes, of the original combinator graph, but by the maximum number of nodes required during reduction. As recursive expansion of functions is common in CGR, we expect that the maximum number of PEs/graph nodes required for a real application may be as high as 100K to 500K [Scheevel], even when concurrent distributed garbage collection is employed as in SPARO. The maximum size of the PE array thus can be very large. For this reason, we may consider a couple of options when implementing SPARO realistically in electronics. As suggested above, we can increase the granularity of the PEs to handle subgraphs instead of single nodes, so that 1K PEs would suffice in handling reductions. However, this would reduce the maximum parallelism that can be expressed in the graph. It would also increase the memory requirements in each processor. As mentioned earlier, the message bandwidth is not expected to change much from that in SPARO since the messages will be of greater length but they may be generated less frequently.

For purposes of solving the PE interconnection problem, we can still derive a major benefit by solving the general message passing problem that features the same bandwidth as that of the messages in SPARO. We will therefore isolate the exact processing nature in the architecture used from the specification and requirements of the network, and focus on achieving a high throughput of messages between PEs in a generic parallel processing environment that uses message passing.

3. 3 SEN Schemes

Based on the previous discussions, we can enumerate a number of options in designing a high-speed SEN that uses optics. These are:

1) serial electronic exchange and parallel optical shuffle and data transfers
2) serial optical exchange and shuffle
3) parallel optical exchange and shuffle

We use the term parallel to include both fully parallel and quasi-parallel data transfers. This broader definition is employed since it is not certain that fully parallel (100 bits) optical data transfers (at the board level) may be possible for a large number of processors. The actual physical size and partitioning of the PEs will dictate the level of parallelism in the message transfer. An example of such (quasi-) parallelism would be to use 25 signal lines (encoded in fewer lines or non-coded data in 25 channels) to transfer the message in four periods. Parallel message transfer in purely electronic computers is not considered because of the complexity of size and packaging.

In option 2 where serial exchange and shuffle are done, the SEN has to be operated 100 times or faster than the speed at which the PEs operate. This is not true for options 1 and 3, as we shall see. Clearly, given the maturity of high-speed optical switching devices, options 2 and 3 are the most difficult and ambitious. We examine the serial scheme (option 2) first.

3.4 Serial optical SEN (Option 2)

The messages are assumed to be generated within the PE and stored in the output buffer (OB) (Figure 2). Each PE has a laser diode array integrated on-chip to implement the 4 handshake lines and the data line to serially stream out the message data once the network acknowledges the PE request for transferring a message (Figure 8 shows a single PE and the SEN connection). Each processor has access to two clocks: the first for the electronic circuitry and the second faster one to clock the OB, IB, and the diode array. It is assumed that both clocks are distributed optically on the chip as well as on the board. The faster clock is assumed to be almost 100 times faster than the slower one. In such a case, the network cycle time is as long as that of the processor. Effectively, the processor and network operate at the same speed to maximize throughput. Note that since the PE can be assumed to load the OB in parallel, there is no delay in moving the message within the PE. The faster operation of the optical SE assumes that the control and exchange can be operated at the higher speed. Thus, if the PEs are high performance processors designed to run

on a clock of 50 MHz (100 MHz), the network clock must operate at 5 GHz (10 GHz). The complete network cycle is then about 20 ns (10 ns), although all switching within the network occurs with a delay of 200 ps (100 ps). To avoid synchronizing problems, the slower clock would be derived from the faster clock.

There are some obvious technical obstacles to the proposal. First, the PE chip has to integrate the high-speed buffers and the laser diode array. We require, at the least, one high-speed buffer, instead of separate OB and IB, which communicates with the optical network. While the laser diode array and the buffers are required to be implemented in GaAs, the rest of the circuitry, the processor and its interface to the external world, would be in Si (ECL) (Figure 8). This is essential since ECL and bipolar have much higher levels of integration for a full-scale processor design than GaAs. Using today's integration capabilities, separate (Si and GaAs) dies can be separately optimized and integrated ona single package. While placing a single laser diode on the package is not a problem implementation is not a problem, integrating a large number of such diodes at such speeds on a single package is challenging. The problems are mainly caused by thermal coupling between the lasers, as well as the minimal size each laser occupies. The limited number of lasers that can be integrated in a package has more impact on the number of message lines (and the number of PEs) that can be put on a chip. If lower speed are used to transmit data out of the PEs, a higher degree of parallelism is possible.

Second, the network must have optical switches and buffers that can implement the control and the exchange at the higher speed. Specifically, we need to be able to implement both logic gates and buffer elements in integrated optics. This is the bigger technical challenge. Currently, we are evaluating the smart exchange switch schemes presented in Table 3.

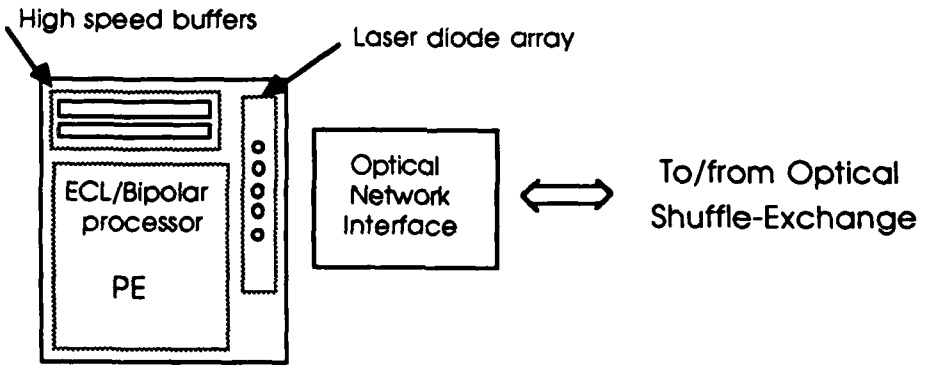

Figure 8. PE and optical SEN connection

3.5 Parallel Optical SEN (Option 3)

The serial optical SEN can be an advantage over an electronic SEN if it switches at a speed that is 100 or more times faster than the electronic PEs. The parallel optical SEN faces the same technical difficulties as the serial SEN in that the switching and the control must be implemented optically. However, because messages are to be transferred in parallel, the speed requirement of the optical logic is much less severe.

If 20 message bits could be sent simultaneously from the PE to the SEN, the switching speed requirement for the optical logic is 20 times lower than the serial SEN case. The difficulty in this scheme, besides those that exist in option 2, is in the limits of optoelectronic integration.

3.6 Electronic logic parallel optical shuffle (Option 1)

This represents the least ambitious of all three options in that it requires no optical logic or buffers. The optical portion of the SEN is in increasing the bandwidth of the network as well as in implementing the shuffle. Figure 9 shows the schematic layout for this SEN scheme.

The PE arrays (single or multiple boards) are connected to the SEN by fiber optic links. Each PE contains a set of laser diodes that are operate only at a specific frequency. A number of these diodes, say 20, are integrated on the PE such that by using a combination of frequency and polarization encoding, all 100 message bits are encoded and transmitted in parallel in a fiber in 3 to 4 clock cycles. This clock can be the same as that of the PE, say 50 to 100 MHz. The limit on the amount of parallelism is specified by the number laser diodes that can be integrated on-chip.

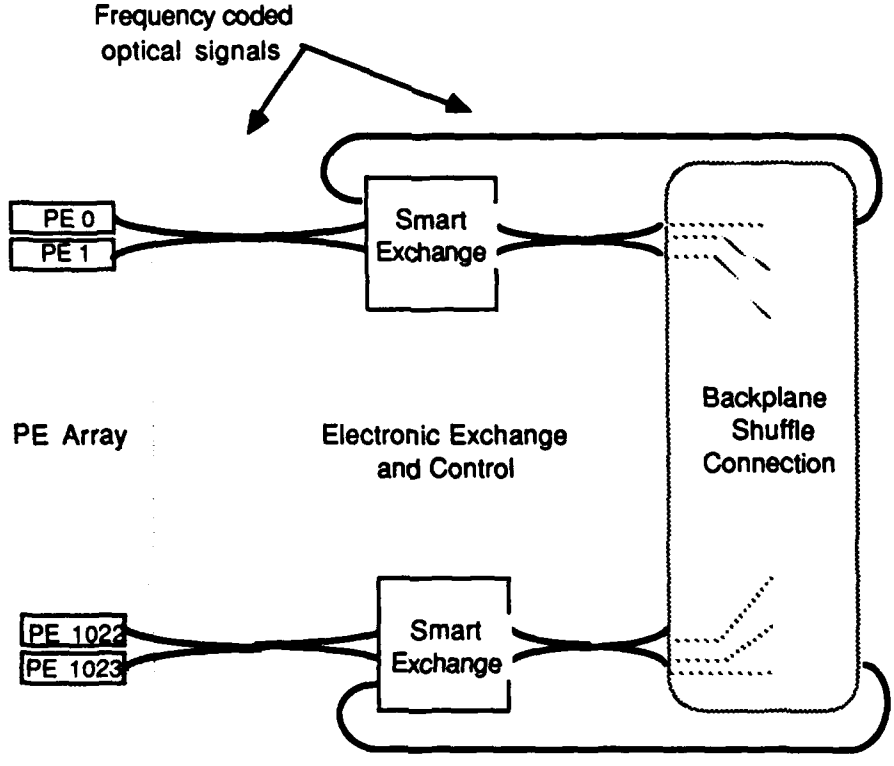

Figure 9. Serial electronic exchange parallel optical shuffle scheme

The smart exchange switches are implemented in electronics. Each exchange module contains electrooptic detectors that detect specific frequencies and polarizations of the incoming message lines. The output of the exchange modules are shuffled and fed back to the network and the PEs. The perfect shuffle can be

implemented by physically connecting the fiber optic cables or by using a holographic optical element (HOE) as discussed later.

While this scheme is the least ambitious, it is more easily implementable than the other schemes in the near-term. It also provides a migration path to a more optical system by replacing the smart exchange switches in optics at a later stage if the optical logic does not provide the desired performance and integration.

3.6.1 Alternate Partitioning in Option 1

The scheme presented in Figure 9 appears to partition naturally in vertical slices, that is, a set of PEs on one board, a set of exchange and control logic on another board, and the shuffle connection at the edge of the exchange and logic board. An alternate means of partitioning the PE array and the SEN might be more attractive in terms of implementation. Consider partitioning the complete architecture into horizontal slices, which are placed on separate boards. Each board is a slice of the architecture consisting of an even number of PEs and their corresponding exchange and control stages. A number of boards, depending on the total number of PEs in the architecture and the number of PEs that fit on a board, are connected by the shuffle connection. The advantage in this scheme is that it avoids routing the message wires between the PEs and the exchange switches across boards. Instead of using optical connections, high density interconnections have to be done on the board. The off-board connection density while limited by the packaging technology, could be improved by the use of optical fiber lines or even one-dimensional SLMs (*low-power* (25 nJ), *small sized* (100um) 1-D Si/PLZT SLMs with BW of 3 MHz have been demonstarted at UCSD).

Based on the different options in physically configuring the processor-network architecture, we believe the above option of partitioning the PEs into different boards to be the most practical. The problem of designing the optical SEN is then redefined as one of distributing the shuffle across multiple boards and designing the optical exchange switch.

The interboard shuffle will be specificed by the number of channels on each board. This is determined by the off-board connection or wire density. The actual off-board wire density required is dictated by the number of PEs placed on a board. The number of PEs placed on each board in turn is determined by the technology used to construct the board as well as the nature of board connectors used. We examine these limits in the next subsection.

3.7 Density of Board Level interconnects

The interconnect density on the board depends on the nature of material used for the board and the type of connectors to communicate off the board. Table 4 shows the possible interconnect density for different materials and technologies. Of the four technologies mentioned, only the standard edge connectors are available off-shelf. The new button board technology (using 4 mil buttons) from TRW which connects boards on the surface is the most promising high-density electronic

interconnect technology possible. At present we do not have information on the upper bound on the number of I/O points on a board that can be connected using this technology. The two off-board optical interconnects techniques are based on optical fibers and waveguides. Of these, the fiber optics technology is more mature. Custom fibers are usually 125 um thick with bend radius not exceeding 2 inches. Waveguides, of dimensions less than 10 um and 10 um spacing, provide the highest density in optical connectivity on or off-board (if alignment problems are solved). The use of waveguides, grown on polyimide substrates at Honeywell, as board level connections is still experimental. Because of their immunity to cross-talk and their non-interactivity, waveguides can be packed at much greater densities than electronic connections.

The different interconnect technologies can be evaluated not only on the basis of density of interconnects but also on their bandwidths. We have therefore listed the limiting speed of operations of each interconnect in Table 4. Complete information on the button board operation is not available yet and is being currently compiled. The advantage in the optical techniques would be their higher available bandwidth.

Type of board	Density/inch	Speed	Comments
PVC/Edge connector	40	200 MHz	Standard/custom
Button Board	170	150 MHz (?)	4 mil buttons
Optical fibers	2000	Source limited	Custom implementation
Polyimide waveguides	12000	Source limited	Experimental

Table 4. Board level interconnect density

3.8 Connectivity Requirements of Some Network Topologies

Since the wiring connectivity requirements of the shuffle exchange appears to be significant, it motivates us to examine other possible or popular interconnection schemes and compare their connectivity requirements. We can show that the wiring complexity across boards, for a multiboard system, of the shuffle-exchange is no worse than that of other interconnection architectures such as the hypercube and the crossbar when parallel message transfers are considered.

Consider, for example, a large-scale parallel architecture consisting of NM PEs distributed across M boards (or clusters, in the general case) communicating via messages. Thus each board has N PEs that need to communicate to other PEs on its board as well as others. Since we are concerned with interconnection requirements across board boundaries for different interconnection architectures, we will not consider the on-board connections that can be done by board-level routing. We will examine the total I/O channels required per board for the same density of PEs on a board for different interconnections.

Hypercube Interboard Connectivity

Examine the hypercube first. Since there are a total of NM (N and M are necessarily powers of 2) PEs, the dimension of the hypercube is $D = \log_2 NM$. Thus each PE has both input and output connections to D other PEs.

If each PE communicates a B-bit data packet or message, then each PE requires DB bits for each input or output connection. To estimate how many of the D PEs are on the same board as the source PE, we have to consider the how the PEs are partitioned.

The best case, that is, when the least number of connections are required outside the board, partition occurs when each board of the N PEs form their own smaller hypercube and D is minimum or 2. In such a case, each PE on a board is connected to $\log_2 N$ PEs on the board and to only 1 PE on the second board. In the general case when there are M boards (M >=2), each board contains N PEs in a hypercube and the number off-board unidirectional PE connections per PE in the best case is $2(\log_2 D - \log_2 N)$ or $2\log_2 M$.

Thus, the number channels required per PE in the board is $2B\log_2 M$.

The total number of channels for each board is $2NB\log_2 M$.

Crossbar Interboard Connectivity

The cross bar connection for connecting a massively parallel PE array does not really make sense since all PEs can talk to each other. However, for purposes of comparison and completeness, we will examine this interconnection architecture. Each PE has to be connected all others. In the multiboard case, each PE has to be connected to all 2N(M -1) off-board PEs, besides the N - 1 on-board PEs.

Thus, the number (unidirectional) channels required per PE in the board is 2BN(M - 1).

The total number of channels for each board is $2BN^2(M - 1)$.

Shuffle-Exchange Connectivity

The computation of the I/O channels required for the shuffle-exchange is relatively simple since each PE has a fixed fanin and fanout of 1. However, in a multiboard situation the partitioning of the processors determines the number of interboard connections.

In the best case, M=2 and the 2N processors can be split up such that only N/2 PEs on each board require off-board connections to the other board. The rest of the PEs can

be connected by the shuffle-exchange on-board. This is because the shuffle connection is bisymmetric (that is, the shuffle connections for the lower group of N PEs are mirror image of the connections of the upper group of N PEs) and half of each group, that is, N/2 PEs, is connected to half of the other group. This can be verified by examining the relation that describes the shuffle permutation S(i) of the ith PE where $0 <= i <= n-1$,

$$S(i) = (2i + \lfloor 2i/n \rfloor) \bmod n$$

The total board I/O required for M=2 is therefore 2BN/2 or BN since there are two connections (input and output) for each of the N/2 PEs connected to offboard PEs.

However, the SEN is not modular, so when M is increased beyond 2, each PE in the worst case partitioning may require a shuffle connection to a PE off-board. In such a case, each PE has I/O connections to two other PEs off-board, one for input and the other for output.

Thus, the worst case number channels required per PE in the board is 2B.

The total number of channels for each board in the worst case is 2BN.

Interconnection Topology	I/O per PE	I/O per Board	Normalized I/O per Board	Board I/O (N = 128, M = 8, B =100)
Hypercube	$2B \log_2 M$	$2NB \log_2 M$	$\log_2 M$	76.8 K
Crossbar	$2BN(M - 1)$	$2BN^2(M - 1)$	$N(M - 1)$	22.4 M
SEN (M = 2)	B	BN	-	-
(M > 2, worst case)	2B	2BN	1	25 K

Table 5. Interboard I/O requirements for different interconnection networks assuming parallel message transfer

Table 5 summarizes the I/O channel requirements for an N PE board where M boards contain a total of NM PEs. In the same table we also show the board I/ requirements for NM = 1024, M = 8 (assuming 128 PEs per board), and B = 100. Note that the columns marked as I/O per PE or I/O per board do not reflect physical fanout but rather the required connectivity. This is because, as in the hypercube operation, the PEs do not operate in a broadcast mode but rather selectively talk to individual PEs at any one time.

Figure 10 shows how the board-level I/O increases as a function of the number of boards for conservative values of N and B (N = 16 and B = 32). Figure 11 and 12 show graphically the total I/O requirements as a function of the number of processors for two different sets of values of message width B and the number of boards M. In each graph we have also provided two reference lines representing the total board I/O possible in two different technologies, button boards and optical fiber interconnects, assuming that a large 18" X 15" board is used. Note that since button

boards have been designed for a maximum of 2000 buttons a 8" X 6" board only, we have extrapolated that figure and assumed that 5000 buttons can be placed on the larger board. In case of optical fibers, we have assumed that they are used only on one edge of the board, and not on the complete periphery like the buttons on the button board. From size and spacing considerations, 36,000 optical fibers can be fitted on the 18" side of the board. The figure is even better (216,000) if waveguide connections can be used on the edge of the board. As Figure 12 shows, for parallel transfer of messages 100 bits wide, the SEN can be supported up to a total of 300 PEs by the use of button board interconnections. Using fiber, a SEN for 3000 PEs could be supported. Because of higher board interconnections, however, the optical fiber approach can support a hypercube of less than 1000 PEs. Thus for a large number of PEs, optical interconnects appear to hold more promise than available electronic technologies.

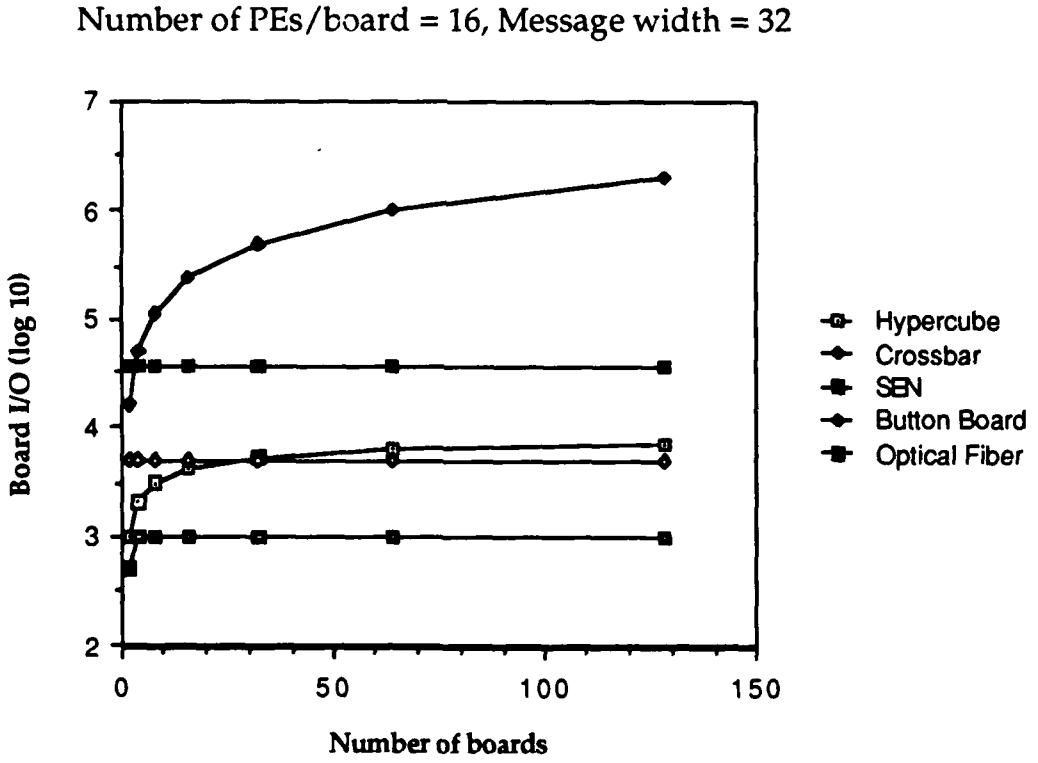

Figure 10. Board-level I/O as a function of the number of boards for B=32, N=16

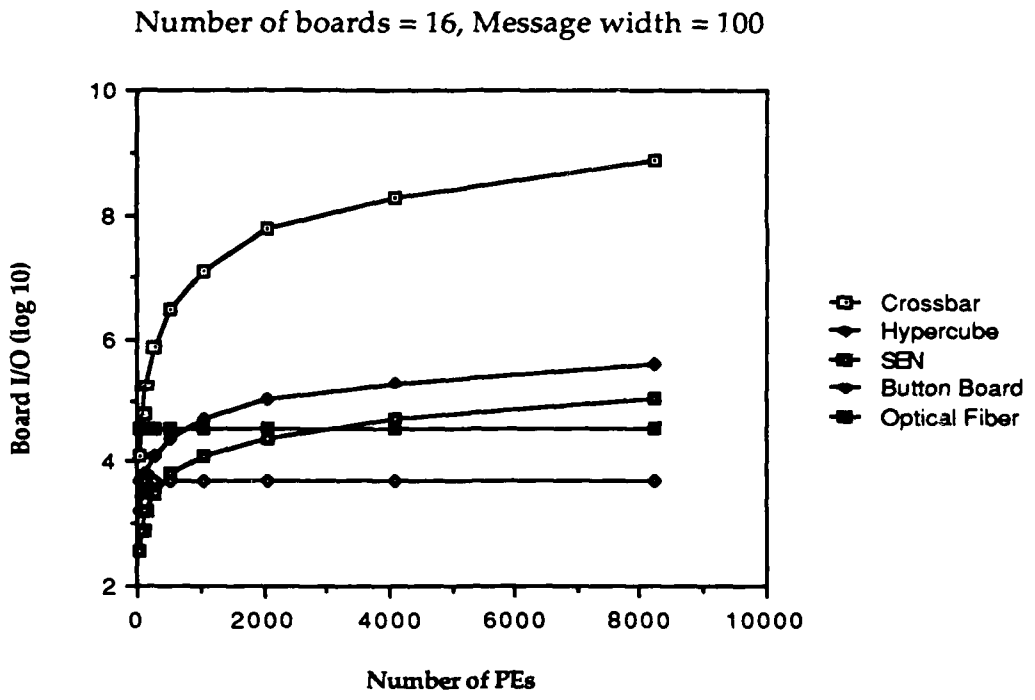

Figure 11. Board-level I/O as a function of the number of PEs/boards for B=10, M=2

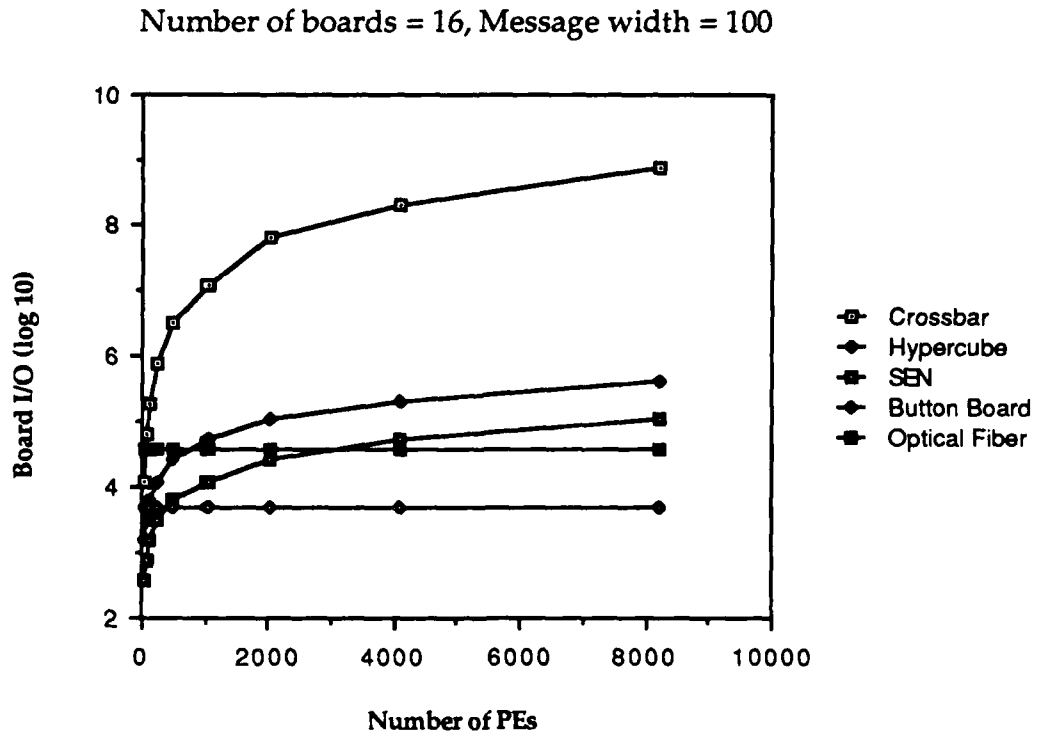

Figure 12. Board-level I/O as a function of the number of PEs/boards for B=100, M=16

Our results on the comparison of the three topologies indicate that the crossbar, because of the number of I/O channels required for parallel data transfer and the associated complexity of implementation, is beyond practical consideration. The hypercube interconnection, although modular, requires higher I/O density than the SE. This is due to the larger fanout, by a factor of $\log_2 M$ over the SEN, of each PE. We note, in fairness, that the limited connectivity of the SEN, implies handling a smaller load, i.e., usually around 25% as mentioned earlier. So larger loads would slow message deliveries down as more conflicts occur. The optimal performance is pegged at a 25% load [Padua and Lawrie], beyond which excessive message conflicts occur. For higher message traffic, we need replicated networks that have the same degree of connectivity as the single-stage SEN but require fanning out and fanning in (same as the order of the replication) of the replicated networks. In the limit, for fully replicated networks, the order of the connectivity is the same as that of the hypercube. However, it is important to note that there is flexibility in the degree of replication so that small scale replication can be used to ensure both lower connectivity and reasonably high performance.

On the other extreme, if MINs (multistage interconnection networks) or multistage SENs are used, the connectivity remains the same but the performance and complexities increase. In fact it has been shown that the SE MINs are optimally cost-effective. However, because of their complexity in implementation, we are restricting our focus on the single-stage SEN which is simpler to implement in optics.

3.9 Nature of Backplane Connections: Guided versus Free-space

Based on our preliminary examination of the optical shuffle, we find at least two broad approaches possible for interconnecting multiple boards which we describe here.

In the first approach, a totally free-space approach such as a hologram (or bulk optics) is used to realize the shuffle permutation mapping of the exchange switch outputs from the edge of the boards to the input of the exchange switches also on the edge of the boards (Figure 13). While the limitation of this approach over the second, where the hologram is parallel to the surface, is that the number of signals (emitters) that can be accessed is significantly smaller, modifications can be made to increase the number of I/O accesses. One instance of such a modification that we are currently examining, is to use bulk imaging techniques on the surface of the board to draw out the signals to the edge and then construct the shuffle at the edge of the board as depicted schematically in Figure 13.

In the second approach, the hologram (or bulk optics) can be placed parallel to the surface of the board and detects signals from vertically emitting diodes on the board (Figure 14). The shuffle mapping is achieved by using waveguides for routing the incoming signals to the proper point on the holographic plane. Although this approach is very attractive because of the larger number of signals (emitters) that can be accessed, large holograms would be required if multiple boards are necessary for the architecture. Another consideration is that the number of connection crossovers

in the backplane is limited by the crosstalk in intersecting waveguides. Using holograms of larger size increases the optical path length and also increases the difficulty in implementing them practically. Therefore, this technique of designing an optical shuffle would appear impractical if the number of boards required is much greater than two. Note that the waveguide connection technique can also be applied to the edge-of-the-board connection shown in Figure 13. This technique is currently under investigation at Honeywell.

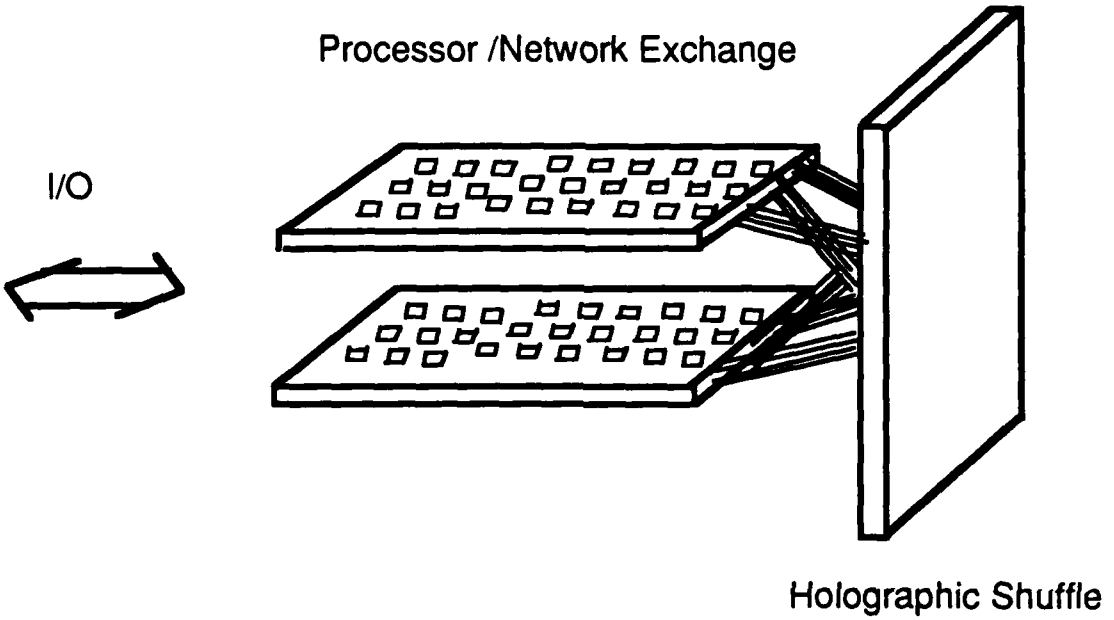

Figure 13. Edge-of-the-Board Optical Shuffle

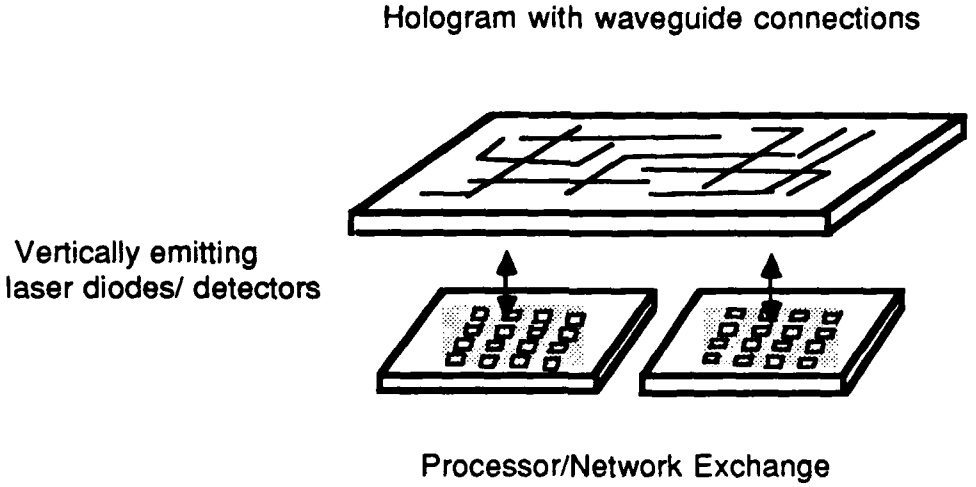

Figure 14. Surface-of-the-Board Optical Shuffle

4. CONCLUSIONS

Based on our detailed analysis of large electronic shuffle-exchange networks and their requirements, we have concluded that a high-density and high-bandwidth shuffle-exchange network is necessary for highly parallel processing. Because of the current limitations of electronics, we believe that optics can be an advantage in implementing the interconnection network.

Our current tasks have two focii. First, and more important, to design an effective method for accomplishing board level shuffles a for a large number of PEs distributed across a large number of boards. This board-level network design problem will surface in massively parallel computing architectures, such as the Connection Machine (CM) or the Ncube, especially when high-speed processors communicate via packet switching. The problem is further compounded when message packets used for communication between PEs are sent in parallel and not serial as in the current implementation of the CM. The second task is the evaluation of an optical implementation of the smart exchange switch. Discrete Boolean implementation using logic gates, as done in electronics, do not make sense for an optical implementation since electronics does better. Using different representation for the logic and data representation is more plausible (as indicated in our preliminary analysis). However, the problem of implementing a large number of such exchange switches for a single board of PEs appears difficult. Further, our analysis reveals that the bottleneck in throughput of the network is in the message transfer rate rather than in the exhcange logic.

REFERENCES

D. H. Lawrie and D. A. Padua, 'Analysis of Message Switching with Shuffle-Exchange in Multiprocessors,' Proc. of the Workshop on Interconnection Networks for Parallel and Distributed Processing, 1980, pp. 116 -123.

S. C. Esener, 'One-dimensional Silicon/PLZT Spatial Light Modulators,' Optical Engineering, May 1987, Vol. 26, No. 5.

A. Guha, 'OSPESE Quarterly Technical Report: September 1, 1987 to November 30, 1987'.

M. Scheevel, 'NORMA: A Graph Reduction Processor,' 1986 ACM Conference on LISP and Functional Programming, August 1986, pp. 212 - 219. Private Communication.